Simulation and Performance Evaluation of the Intel Rate Adaptation Algorithm

Rémy Grünblatt, Isabelle Guérin-Lassous, Olivier Simonin

Simulation and Performance Evaluation of the Intel Rate Adaptation Algorithm

Rémy Grünblatt
remy@grunblatt.org
Univ Lyon, EnsL, UCBL, CNRS, Inria, LIP & CITI, France

Isabelle Guérin-Lassous
isabelle.guerin-lassous@ens-lyon.fr
Univ Lyon, EnsL, UCBL, CNRS, Inria, LIP, France

Olivier Simonin
olivier.simonin@insa-lyon.fr
Univ Lyon, INSA Lyon, Inria, CITI, France

ABSTRACT

With the rise of the complexity of the IEEE 802.11 standard, rate adaptation algorithms have to deal with a large set of values for all the different parameters which impact the network throughput. Simple trial-and-error algorithms can no longer explore solution space in reasonable time and smart solutions are required. Most of the WiFi controllers rely on proprietary code and the used rate adaptation algorithms in these controllers are unknown. Very few WiFi controllers provide their rate adaptation algorithms when they do not rely on the Minstrel-HT algorithm, which is implemented in the Linux kernel. Intel WiFi controllers come with their own rate adaptation algorithms that are implemented in the Intel IwlWifi Linux Driver which is open-source.

In this paper, we have reverse-engineered the Intel rate adaptation mechanism from the source code of the IwlWifi Linux driver, and we give, in a comprehensive form, the underlying rate adaptation algorithm named Iwl-Mvm-Rs. We describe the different mechanisms used to seek the best throughput adapted to the network conditions. We have also implemented the Iwl-Mvm-Rs algorithm in the ns-3 simulator. Thanks to this implementation, we can evaluate the performance of Iwl-Mvm-Rs in different scenarios (static and with mobility, with and without fast fading). We also compare the performances of Iwl-Mvm-Rs with the ones of Minstrel-HT and IdealWifi, also implemented in the ns-3 simulator.

ACM Reference Format:
Rémy Grünblatt, Isabelle Guérin-Lassous, and Olivier Simonin. 2019. Simulation and Performance Evaluation of the Intel Rate Adaptation Algorithm. In *22nd Int'l ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '19), November 25–29, 2019, Miami Beach, FL, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3345768.3355921

1 INTRODUCTION

One of the requirements of the IEEE 802.11 standard (also known as the *WiFi* technology) is the ability to handle portable and mobile stations, required in applications such as mobile robot fleets. For this purpose, the IEEE 802.11 physical layer (*PHY*) defines many transmission features that can be chosen and combined in order to ensure the good receipt of data, notwithstanding changes in the transmission channel caused by mobility or propagation effects. Examples of such transmission features are the transmission modulation and coding scheme, the use of space-time block coding, the guard interval length, the channel width or the number of spatial streams. The IEEE 802.11 standard defines which combinations of transmission features are allowed and forbidden, but it does not enforce any behaviour regarding how these features should be chosen, letting each station (*STA*) in charge of deciding its own transmission features. The combination of these different transmission parameters corresponds to a transmission rate. The algorithms automatically choosing these parameters, called **rate adaptation algorithms** (*RAAs*), may optimize transmissions with regards to different metrics such as robustness, throughput [13] or energy consumption [14]. With the rise of the complexity of the physical layer of the IEEE 802.11 standard, the size of the search space of the different transmission parameters has increased by two orders of magnitude, making the design of efficient rate adaptation algorithms an active field of research. Indeed, simple trial-and-error algorithms can no longer explore solution space in reasonable time.

A large number of rate adaptation algorithms have been proposed in the literature. It is difficult to know which solutions are effectively used in real WiFi products as a large part of the code of WiFi drivers is proprietary and not accessible. Some WiFi Linux drivers, like the Soft-MAC drivers, have the advantage of using the Media Access Control management entity implemented inside the Linux kernel in the mac80211 component, allowing an open-source implementation to drive multiple pieces of hardware. Currently, two rate adaptation algorithms are implemented in mac80211, namely Minstrel [4] and Minstrel HT [6]. Only a handful of Soft-MAC drivers use their own rate control algorithms and not the ones implemented inside the mac80211 component. This is for instance the case of the Intel wireless main driver, named *IwlWiFi*, that uses its own rate adaptation algorithms Iwl-Agn-Rs and Iwl-Mvm-Rs.

As far as we know, no study in the literature describes the rate adaptation algorithms used by Intel and their performance. This absence of study is surprising for several reasons. Rate adaptation plays an important part in 802.11 performances, such as throughput or delay and these performances have an impact on the communication quality. Intel WiFi controllers are common, as they are used in many laptops, but also in more exotic devices, such as the Intel Aero Ready-to-Fly UAVs (Unmanned Aerial Vehicle). The mobility capabilities of these devices lead to greater dynamics in radio conditions, and therefore introduce a need for a suitable and efficient rate adaptation algorithm.

In this paper, we study a rate adaptation algorithm used in real WiFi products. Our main contributions are the following:

MSWiM '19, November 25–29, 2019, Miami Beach, FL, USA

ACM ISBN 978-1-4503-6904-6/19/11…$15.00
https://doi.org/10.1145/3345768.3355921

- We present the RAA used in recent Intel hardware wireless network interface controllers (WNIC), Iwl-Mvm-Rs, which has been reversed-engineered from the Intel IwlWifi Linux driver source code. This is the first study that gives and explains the rate adaptation algorithm used by Intel in a comprehensive form.
- We compare and analyze the performances of the Iwl-Mvm-Rs, IdealWifi and Minstrel-Ht RAAs with regard to the mobility of stations using simulation. In particular, we show that Iwl-Mvm-Rs is better at dealing with mobility, in the tested scenarios, than the two other algorithms. We use the network simulator ns-3 because this is the most up-to-date simulator concerning the WiFi technology and because several RAAs are already implemented in NS-3.
- We provide an open source implementation of the Iwl-Mvm-Rs algorithm in the network simulator NS-3.

The paper is organized as follows. Section 2 gives the main new features that have been introduced in the IEEE 802.11n and 802.11ac versions and that are used in the rate adaptation algorithms under study in this paper. A brief state-of-the-art on RAAs for WiFi networks is also provided. In Section 3 we describe the rate adaptation algorithm Iwl-Mvm-Rs we have reverse-engineered from the Intel IwlWifi Linux Driver source code. We give a short description on the implementation of this algorithm in the ns-3 simulator. Then, in Section 4, we compare, by simulation the performance of Iwl-Mvm-Rs with two other algorithms. Finally, we conclude in Section 5.

2 BACKGROUND AND RELATED WORK

2.1 WiFi Networks

Most of the current WiFi products use the IEEE 802.11n and 802.11ac standards[1]. Several new approaches have been introduced to improve performances compared to the first IEEE 802.11 modes. At the physical layer, stations can be equipped with multiple antennas and MIMO (Multiple-Input Multiple-Output) is used to increase transmission rates and reliability by transmitting multiple spatial streams simultaneously and by exploiting the spatial diversity. The maximum coding rate is increased from 3/4 to 5/6 and a short guard interval of 400 ns is introduced to improve spectral efficiency. In addition, it is possible to aggregate several channels in order to transmit on larger channel width. 802.11n offers the possibility to transmit on 40MHz channels, while the use of 80MHz or 160MHz channels is possible with 802.11ac.

The combination of the modulation type and the coding rate is represented by a Modulation and Coding Scheme index (MCS)[1]. Coupled to the channel width, the number of spatial streams and the guard interval length, these parameters define a physical transmission rates.

At the MAC layer, the frame aggregation mechanism aggregates multiple data frames before transmission. Two aggregation mechanisms have been proposed: A-MSDU (Aggregate MAC Service Data Unit) aggregates several MSDUs, carried within a single data MAC protocol data unit (MPDU), and A-MPDU (Aggregate MAC Protocol Data Unit) aggregates several MPDUs carried within a single physical layer convergence procedure service data unit (PSDU). While not changing the physical transmission rate, these schemes allow for less overhead and therefore higher throughputs.

2.2 Rate Adaptation Algorithms

Wifi offers a large range of transmission parameters. Taking into account guard interval duration, channel bandwidth, MCS index and the number of spatial streams, 802.11ac supports 312 different transmission combinations resulting in 122 different rates, while 802.11n supports 130 combinations and 70 transmissions rates.

The problem of rate adaptation for WiFi networks has been studied for two decades, since the IEEE 802.11a/b amendments that introduce multiple transmission rates. For 802.11n/ac standards, various solutions have been proposed in the literature[3, 5, 9–11]. Among the recent works on the subject, one can note the works in [8, 12]. The solution described in [8] is one of the first solutions to consider the mobility and its impact on the transmission rate adaptation. The solution is based on an adaptive learning mechanism. Even if this solution shows promising results, it is not yet implemented in WiFi interfaces.

3 THE INTEL RATE ADAPTATION ALGORITHM

Intel wireless chips use the IwlWiFi driver. It comes with its own rate adaptation algorithms: Iwl-Agn-Rs and Iwl-Mvm-Rs, the former being un-maintained, and limited to 802.11n hardware, while the latter is being used with 802.11ac compatible Mvm hardware. Therefore, we focus our study on the algorithm Iwl-Mvm-Rs. As far as we know, there is no study that describes this algorithm and its performances. Only the code is open. To come with a comprehensive algorithm, we have reverse-engineered the code provided by Intel [2]. The IwlWiFi driver has approximately 72.000 source lines of code, 40% of which are used by the Mvm hardware. The RAA is located in the `rs.c` file which has approximately 3200 source lines of code. In the following subsection, we provide a description of the Iwl-Mvm-Rs algorithm.

3.1 Algorithm Description

Iwl-Mvm-Rs takes care of managing the Modulation and Coding Scheme (MCS) index, but also whether to transmit in a legacy mode (802.11a or 802.11g) or in a non-legacy mode (802.11n or 802.11ac) in a SISO or MIMO way. It chooses which antenna or subset of antennas to transmit with, and whether a Short Guard Interval (Sgi) or a Long Guard Interval (Lgi) is used. It also decides when to enable frame aggregation, and can do transmission power control under certain conditions, but this has not been studied in this paper.

Iwl-Mvm-Rs has two main components: MCS Scaling[2] and Column Scaling. MCS Scaling tries to maximize the throughput by only changing the MCS, while Column Scaling tries to find a better *column*, which is a combination of *mode* (legacy, SISO, MIMO), *guard interval*, and *antenna configuration* parameters. The algorithm starts with the lowest transmission rate, which has the best reliability, and interleaves MCS Scaling phases and Column Scaling phases,

[1]For 802.11n, the MCS index is used to encode the number of spatial streams, but we use it here as defined by the 802.11ac standard.

[2]Even if the MCS concept does not exist until 802.11n, this term is used as a handy shortcut to refer to both MCS and data rates.

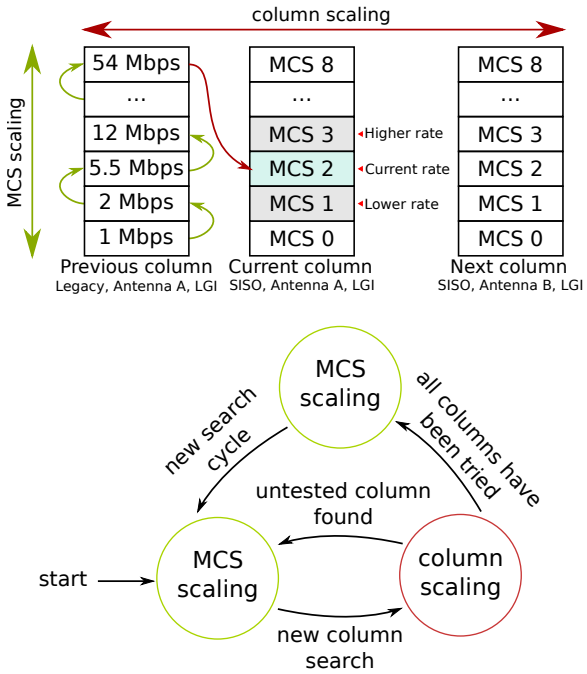

Figure 1: (Top) Example of sequence of decisions made by the RAA Iwl-Mvm-Rs; (Bottom) Flowchart of the different states of the RAA Iwl-Mvm-Rs.

forming a "search cycle". Column Scaling starts when the MCS Scaling phase chooses not to change MCS. The alternation of MCS Scaling and Column Scaling continues until all the columns have been tried, which means the end of the search cycle and the MCS scaling phase runs until the beginning of a new search cycle. Figure 1 sums up the different steps of the algorithm, steps described in more details hereafter.

MCS Scaling. The MCS scaling algorithm can take one of the following decisions: lowering the MCS index, raising the MCS index, or keep using the current MCS index. The decision is made in a deterministic manner, according to the maximum theoretical throughput and the measured throughput of adjacent MCS indexes as well as the measured throughput of the current MCS. It therefore prevents from switching to a MCS implying a theoretical throughput lower than the current measured throughput, or to a MCS whose measured throughput is worse than the current one, for example.

The theoretical throughput is hardcoded into tables for each mode (legacy, SISO, MIMO), each MCS index, and for the four possible guard interval and aggregation parameters (Sgi, Lgi, Sgi+Agg, Lgi+Agg). The measured throughput for each MCS index is computed by multiplying the success ratio of up to the last 62 frame transmissions at this MCS with the theoretical throughput when using this MCS. At least 8 successful transmissions or 3 failed transmissions are required to compute the success ratio: no decisions are taken until these thresholds are met. The decisions are the following:

(1) if the success ratio is too small (< 15%) or the measured throughput is zero, **decrease** the MCS index;
(2) else:
  (a) if the measured throughputs with the lower and higher adjacent MCS indexes are unknown;
  (b) or the measured throughput with the lower adjacent MCS index is worse and the measured throughput with the higher adjacent MCS index is unknown;
  (c) or the measured throughput with the higher adjacent MCS index is better;
  increase the MCS index;
(3) else, if the measured throughputs with the lower and higher adjacent MCS indexes are worse, **maintain** the MCS index;
(4) else, if the success ratio is lower than 85% and the lower adjacent MCS index throughput can theoretically beat the current measured throughput, and:
  (a) if the measured throughput with the lower adjacent MCS index is better;
  (b) or the measured throughput with the lower adjacent MCS index is unknown;
  decrease the MCS index;
(5) else, **maintain** the MCS index.

Column Scaling. Each column has a set of "next columns" that the driver will try if they can theoretically beat the current measured throughput (by looking at the maximum theoretical throughput of the columns), and if they have not been already tested during the search cycle. When trying a new column, if the measured throughput in this column is better than the throughput in the previous one, the RAA keeps using it. Otherwise the column is marked and the RAA reverts back to the old column, into the MCS scaling phase.

The initial starting MCS index in the new column is chosen according to the success ratio: if it is high enough (more than 85%), the smaller MCS index whose theoretical throughput is higher than the current theoretical throughput is chosen. Otherwise, the smaller MCS index whose theoretical throughput is higher than the current measured throughput is chosen. After a column switch, the measured throughput of the previous column are dropped.

New Search Cycle. After the end of a search cycle, the algorithm does MCS Scaling until the start of a new search cycle. This start is triggered when:

(1) too many frames have failed (160 in legacy, 400 otherwise) since the beginning of the previous cycle;
(2) too many frames have succeeded (400 in legacy, 4500 otherwise) since the beginning of the previous cycle;
(3) too much time has been spent after the end of the previous search cycle. The maximum time between two consecutive cycles is set to 5 seconds.

Aggregation. A-MSDU is disabled in non-legacy mode when the MCS index is smaller than 5 (corresponding to a modulation type of 16-QAM, QPSK or BPSK), or when the MCS index is decreased. A-MPDU is enabled on a per-hardware queue basis, depending on the traffic identifier of the data.

Retry Chain. Frames are re-transmitted up to 15 times. The first re-transmission uses the same transmissions parameters, the next

4 re-transmissions use the lower two MCS indexes in the same column, and the ones after change the used column and use decreasing MCS indexes and alternating antennas.

Bandwidth. The RAA uses the maximum bandwidth supported by the standard it uses and by the STA it communicates with. For legacy rates, it therefore uses 20Mhz, but it may use 40Mhz, 80Mhz or 160Mhz for non-legacy rates.

3.2 Simulation and Validation

In order to study the Iwl-Mvm-Rs algorithm and evaluate its performance in various scenarios, we have implemented it in the ns-3 simulator. The source code is available in [7].

ns-3 code. The algorithm has been implemented as a *VHT* (Very High Throughput) low-latency WifiManager in ns-3 . As the language used to implement the manager algorithms in ns-3 is C++, one could have blindly translated the rate adaptation algorithm of the driver code from C to C++ and simulate the behaviour of Iwl-Mvm-Rs. Still, many parts of the driver code have only housekeeping functions for the underlying hardware, such as catching unexpected bugs or re-synchronizing the state of the rate adaptation algorithm (that runs in the CPU) with the state of the hardware (the WNIC). As these behaviours should not happen in a simulator, the RAA in the simulator has been re-implemented using the skeleton of the RAA in the Intel driver, but is not a one-to-one correspondence. The number of source lines of code of the simulated RAA is thus cut by two-thirds with regards to the driver code.

The simulation covers most of the algorithm but ignores some part of the original RAA for the sake of simplicity. First, it blindly enables A-MPDU aggregation when possible, as we assume no real-time traffic will be sent. Then, the retry chain does not use decreasing transmission rates but instead uses the current transmission rate. As re-transmissions of lost MPDUs in a A-MPDU frame are not made using this retry chain because missing MPDUs are resent as a part of the next A-MPDU, and as the first re-transmission rate used in the retry chain is the original transmission rate, we believe this simplification introduces no major changes in the results of our simulations.

Validation. In order to validate the behaviour of the simulated code, we have compared the decisions of the rate adaptation algorithm with the decisions of a real piece of hardware, the Intel Corporation Wireless 8260 WNIC, in different situations. To get an easy access to the decisions made by the RAA on a Linux system, one can load the IwlWifi Linux kernel module with the option 'debug=0x00100000', which enables debug messages about the rate adaptation process inside the kernel log (accessible using the 'dmesg' command). Multiple patterns observed over-the-air are correctly reproduced in the simulator, but these results are not presented here due to space constraints.

4 PERFORMANCE EVALUATION AND MOBILITY

We use the network simulator ns-3 (version 3.29) to evaluate and compare the performances of Iwl-Mvm-Rs with the Minstrel-Ht and IdealWifi algorithms. We choose Minstrel-Ht because it is implemented in the mac80211 kernel component. Minstrel-Ht and IdealWifi algorithms are also the only algorithms implemented in the simulator supporting VHT, that is to say 802.11ac. There exists a third rate adaptation algorithm in ns-3 supporting VHT, ConstantRate, which is of limited interest as it uses static transmission parameters and therefore does not perform any rate *adaptation*. The ns-3 IdealWifi manager transmits the signal-to-noise (SNR) ratio of each frame using a perfect out-of-band mechanism, to the emitter. The latter then chooses transmission parameters maximizing throughput and maintaining the bit error rate (BER) below 10^{-5}. To measure the adaptability and the responsiveness of these algorithms, we design two simulation scenarios involving node mobility or sudden changes in the communication channel characteristics, as well as a scenario where the transmission conditions do not change, acting as a baseline simulation. All of the devices use 2 antennas supporting 2 spatial streams in transmission and reception (so-called "2x2:2" devices) using a 20Mhz bandwidth and the channel 42 (5210MHz).

Simulations #1 and #2 involve two stations (STA), both running the same RAA, one acting as UDP traffic generator and the other one acting as a sink, and the simulation #3 introduces a relay STA. At $t = 1s$, the traffic generator sends UDP datagrams of size 1420 bytes to the sink, until the end of the simulation, either at the specified data rate, either in saturation.

4.1 Scenario #1 - Fixed distance

In this scenario, the two STAs are static and are separated by a fixed distance. Simulations last 30 seconds and each result is the mean of 5 simulations. The distance is increased with a step of 1m. The simulation uses the log-distance path loss model, described in Equation (1), which models the path loss Pl as a function of the distance d and of γ, an environment-dependent constant called the loss exponent, and the power Pl_0 at distance $d = d_0$.

$$Pl(d) = Pl_0 + 10\gamma \log_{10}(\frac{d}{d_0}) \tag{1}$$

We optionally add a Nakagami-m fast fading loss model to account for the changes in power due to the presence of multiple paths. The expression of the added loss, denoted Pn, at distance d and when incoming power is equal to P is given in Equation. (2).

$$Pn(d, P) = X(m, P/m) \tag{2}$$

with X a realization of the $\mathcal{X}$ Erlang random variable whose density function is:

$$f(x; k, \mu) = \frac{x^{k-1} e^{-\frac{x}{\mu}}}{\mu^k (k-1)!} \quad \text{for } x, \mu \geq 0$$

The parameter m is chosen to be 1.5 for distances smaller than 80m and 0.75 for distances bigger than 80m, which are the default parameters used in the ns-3 model.

The mean throughput (measured at the application level of the sink) with regards to distance is depicted on Figure 2, without fast-fading (on top) and with fast-fading (bottom). Without fast-fading, as no time-varying fading is present, the reception power of the frames remains constant during each simulation. While Minstrel-Ht and Iwl-Mvm-Rs have overall comparable performances, IdealWifi performs significantly worse at distance larger than 15 meters

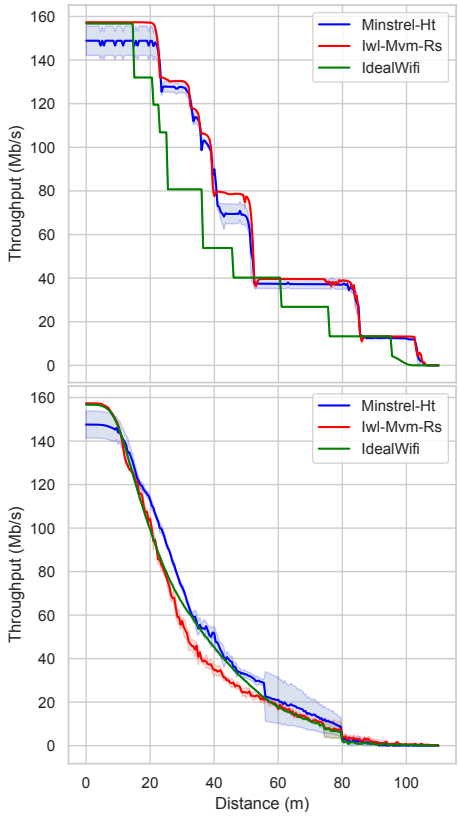

Figure 2: Scenario 1. Throughput as a function of the distance between the source and the sink in the first scenario, without fast-fading (top) and with fast-fading (bottom). Shaded regions represent the standard deviation.

because it does not use transmission rates that would result in a BER bigger than 10^{-5}. By doing so, it does not use transmission parameters that would result in a better throughput. With Nakagami-*m* fading, overall, Minstrel-Ht performs best, and Iwl-Mvm-Rs performs worst. For example, at distance $d = 45$m, throughput is divided by a factor 2.2 for Iwl-Mvm-Rs compared to the case without fast-fading, while it is only divided by a a factor 1.7 for Minstrel-Ht and a factor 1.1 for IdealWifi.

We now comment on the reasons behind the loss of throughput for the Iwl-Mvm-Rs algorithm. The Nakagami-*m* fading is a time varying fading. It means that the reception power for each frame may vary a lot while the position or the mobility of the STAs remains the same, with periods of destructive fading and constructive fading. Looking at the transmission rates used by Iwl-Mvm-Rs, we can observe that overall, the average transmission rate of Iwl-Mvm-Rs is lower than the ones of Minstrel-Ht and IdealWifi. The reasons behind this behaviour are present in the two phases of the Iwl-Mvm-Rs algorithm, the MCS scaling phase and the Column scaling phase. First, in the MCS scaling phase, the algorithm will make its decisions based on at least 3 failed transmissions or 8 successful transmissions. The randomness introduced by the fading may result, locally, in bad performances when these decisions are made, leading the algorithm to stop its exploration and choose a smaller transmission rate instead of climbing the MCS ladder and finding a transmission rate that would result in a better throughput over the long term. At the heart of this behaviour is the asymmetry between the number of failed transmissions and the number of successful transmissions needed to take a decision, as well as the low number of frames required to take a decision. A bigger test window would result in more robust estimations of the potential throughput associated with a given MCS. Then, in the column scaling phase, only a single MCS (corresponding to the smaller MCS index whose theoretical throughput is higher than the current theoretical throughput or the current measured throughput, depending on the success ratio) is tested to decide whether a more in-depth exploration (*i.e.* a MCS scaling phase) should be done in the tested column. As previously, this single test can be very short and coincide with a period of destructive fading, resulting in the whole column being wrongly marked as unsuitable.

Figure 3 confirms the results from Figure 2. It represents, for each manager, the distribution on the transmission rates used by the source to send its frames when the distance between the source and the destination is 45m. Without fading, IdealWifi use a lower transmission rate for a large number of frames compared to Iwl-Mvm-Rs and Minstrel-Ht. These latter two mainly use the same transmission rate, but Iwl-Mvm-Rs sends more frames and sometimes use a higher transmission rate. Table 1 gives the associated mean transmission rate and the success ratio. It shows that without fading Iwl-Mvm-Rs achieves a good trade-off between the mean transmission rate and the success ratio, leading to the highest throughput with this distance. Conversely, Figure 3 shows that, with fading, Iwl-Mvm-Rs mainly uses smaller transmission rates than the other solutions. Minstrel-Ht uses the higher transmission rates for most of the sent frames and thus has the higher mean transmission rate. Table 1 shows that the transmission rates used by Minstrel-Ht also lead to frames losses since its achieved success ratio is around 71%. However the trade-off achieved by this manager is good enough to transmit data with the highest throughput.

Table 1: Scenario 1. Mean transmission rate and success ratio for a distance of 45m.

Solution	Without fading		With fading	
	Mean transmission rate (Mb/s)	Success ratio	Mean transmission rate (Mb/s)	Success ratio
Iwl-Mvm-Rs	86.1	97.5%	38.3	88.5%
Minstrel-HT	80.7	98.1%	74.0	71.5%
IdealWifi	57.0	99.9%	56.4	83.2%

4.2 Scenario #2 - Circular Alternating Walls Shadowing

In this scenario, the source moves in circle around the static sink, at constant speed. The overall distance $d_s = 30m$ between the two STAs does not change during the simulation, but walls are present at distance $d_w = 15m$ for angles θ ranging in $[\pi/4; \pi/2]$,

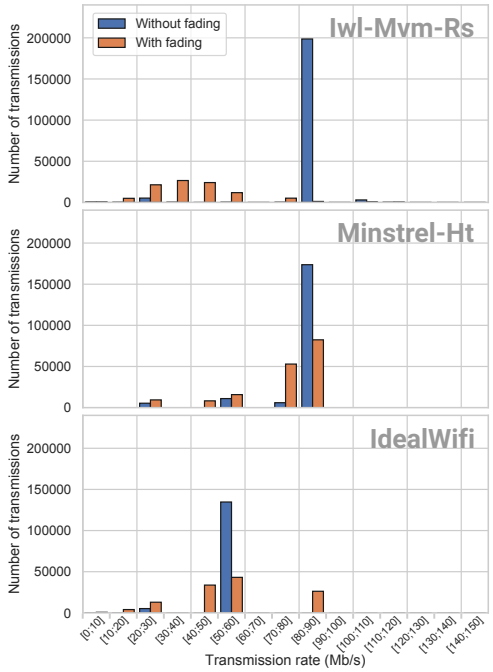

Figure 3: Scenario 1. Distribution of the used transmission rates for the frame sent by the source, without or with fading and for a distance of 45m. Top to bottom: Iwl-Mvm-Rs, Minstrel-Ht, IdealWifi.

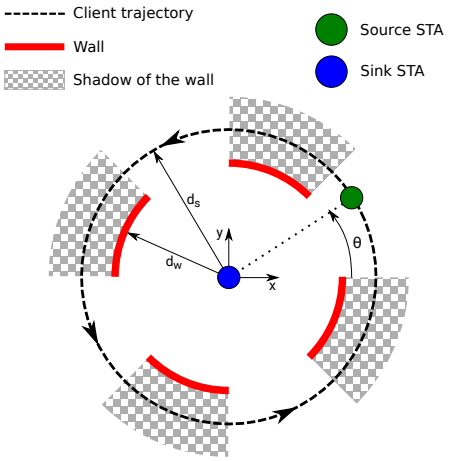

Figure 4: Scenario 2. Top view of the "Circular Alternating Walls" simulation. Walls create shadows leading to sudden changes in the channel between the source and the sink.

$[3\pi/4;\pi]$, $[-3\pi/4;-\pi/2]$ and $[-\pi/4;0]$, as shown on Figure 4. In the "shadows" of the walls, a fixed loss of 5 dBm is added to account for the presence of an obstacle. A log-distance path loss model is used, as well as an optional fast Nakagami-m fading. Each simulation lasts five laps and each result is the mean of 5 simulations.

A perfect rate adaptation algorithm would detect the presence of a wall between the source and the sink without having to send

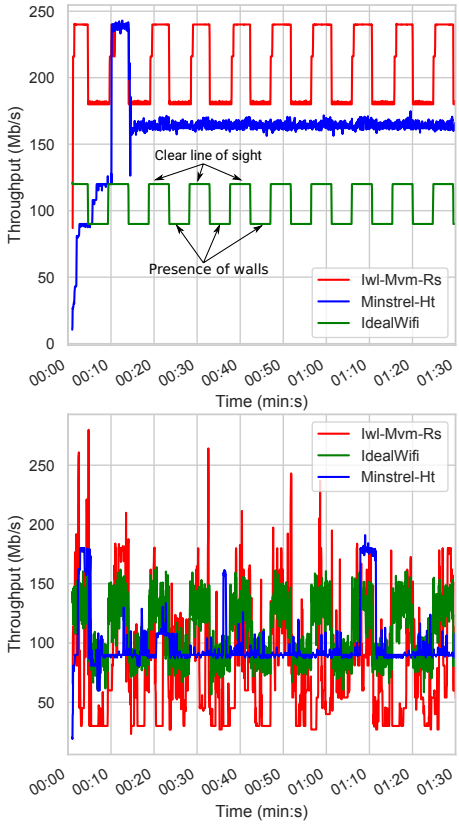

Figure 5: Scenario 2. Evolution of the throughput in the Alternating Wall Scenario for a source throughput of 125 Mb/s and a speed of 5 m/s. Throughput are averaged over periods of 50 ms. Top is without fading, bottom is with fading.

a frame and would change its transmission parameters as soon as needed to counteract the loss induced by shadowing. A slightly less perfect RAA can detect the presence of a wall by detecting changes in its inputs, for example the RSSI, and acts upon these changes to use the best transmission parameters. This is the approach used by IdealWifi, based on the SNR computed at the destination, but this approach is not used in practice for two main reasons. First, it's not realistic to have a perfect out-of-band mechanism for the receiver to send the SNR to the transmitter. Second, it's not always useful to react upon every change in the channel. Indeed, if we consider an hypothetical channel that perfectly transmits one frame out of two and attenuates the other one, then a simple reactive algorithm such as IdealWifi will fail to achieve good performances. As the channel alternates between good and bad transmission conditions and as the RAA uses the previous state of the channel (i.e. an history of 1 frame) to decide which transmission parameters it will use for the next frame, frames with a high transmission rate will be transmitted when the channel is bad, resulting in losses, and frames with low transmission rates will be transmitted when the channel is good, leading to a successful delivery but a loss in throughput. To avoid these pitfalls, real-world RAAs use an history of frame transmission successes and losses and average over this history,

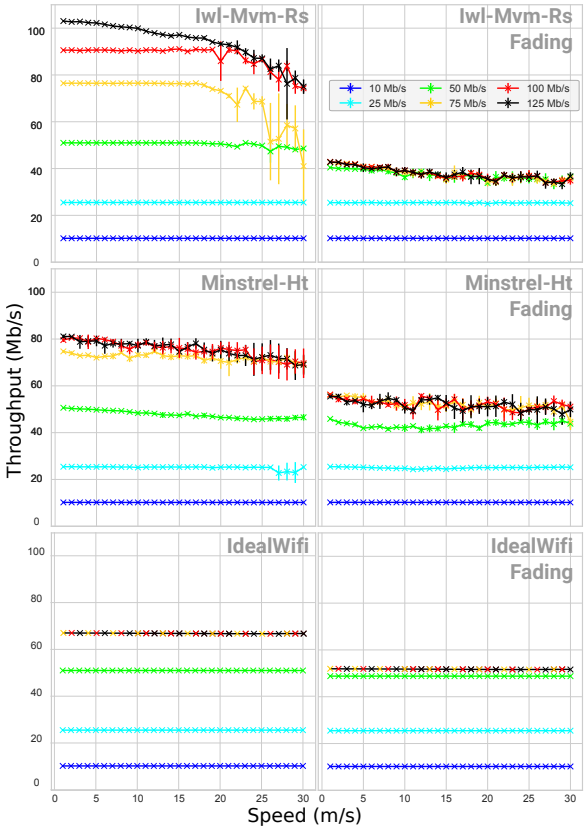

Figure 6: Scenario 2. Evolution of the throughput with regards to the speed and the throughput of the source. Left: without fading, Right: with Nakagami-*m* fading. The vertical bars represent the standard deviation.

using methods like the exponentially weighted moving average of Minstrel or Minstrel-Ht, or a simple average for Iwl-Mvm-Rs. These averages act as low-pass filters and create inertia, which in turns may create unresponsiveness in the decisions of the RAAs.

The goal of the *Circular Alternating Walls* scenario is to test the responsiveness of the RAAs. By making sudden changes in the channel, we can illustrate how well a RAA can react to the presence of a wall or a building, which is important in an urban context. The quicker the RAAs adapt, the best, as they can use the full capacity of the channel. Figure 5 presents the evolution of the throughput in this scenario, for a source throughput of 125 Mb/s and a speed of 5 m/s. Without fading, one can observes that the Minstrel-Ht algorithm is not switching rate when walls are present while Iwl-Mvm-Rs and IdealWifi react to the presence of walls, which illustrates the fact that a RAA might not adapt at all to the presence of walls.

Figure 6 shows the evolution of the throughput with regard to speed for the three managers, with and without fading. One observes that above a certain source throughput, the sink throughput measured at the application level decreases as the speed of the moving source increases for both Minstrel-Ht and Iwl-Mvm-Rs. For source throughput of 10Mb/s and 20Mb/s, the impact of the source

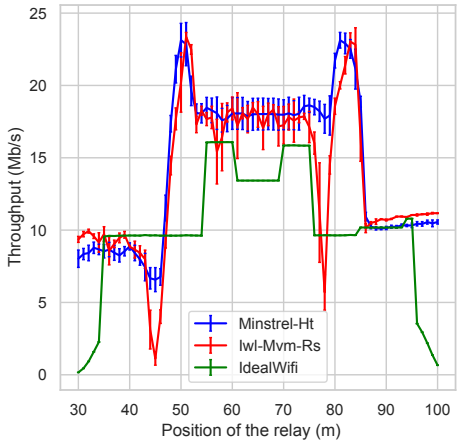

Figure 7: Scenario 3. Evolution of the end-to-end throughput with regards to the relay position. The source is located at d=0m and the sink is located at d=130m.

speed is limited as the same transmission rate is used whether the source is in the shadows or not, because it can accommodate the needs in throughput. At higher source throughput, such as 75Mb/s and above, a small but steady decrease of the throughput with the speed of the source can be observed for Minstrel-Ht. The big differences in performances between Minstrel-Ht and Iwl-Mvm-Rs are explained by the fact the former does not do any adaptation (as shown on Fig. 5) while the latter does.

4.3 Scenario #3 - Two-hop Flow with Relay

In this scenario, one considers three static STAs communicating in a two-hop fashion. The source sends packets to the sink, packets which are forwarded by the relay. The source is positioned at $x = 0$m and the sink is positioned at $x = 130$m, while the relay position changes depending on the scenario. As in the *Fixed distance* scenario (Scenario 1), the source saturates its communication link with UDP datagrams at $t = 1$s until the end of the simulation that lasts 30s. The results in terms of throughput (measured at the application level of the sink) depending on the position of the relay are displayed on Figure 7. We observe a central symmetry in the throughput results, centered on $x = 65$m, which is expected due to the topology of the simulation. When the relay is far from one of the endpoint (source or sink), then the obtained throughput is low since one of the links is of bad quality. As a result, the sink throughput is capped by the throughput of the low quality link.

With all three RAAs, we get the best performance when the relay is not equidistant from the source and the sink. This is explained by threshold effects, as the quality of the channel does not change linearly with the distance, leading to a higher mean channel quality between the source, the relay and the sink at distance 72m than at 65m. The fact that the IdealWifi manager exhibits this behaviour, while solely basing its decisions on SNR, is another argument for this explanation. In Table 2, one compares the mean transmission rate and the success ratio for the source-relay and the relay-sink links. for the two links, when the relay is located at 65m and 82m from the source.

Table 2: Scenario 3. Mean transmission rate and success ratio.

Solution	$d = 65$		$d = 82$	
	Mean transmission rate (Mb/s)	Success ratio	Mean transmission rate (Mb/s)	Success ratio
Iwl-Mvm-Rs				
source - relay	42.6	94.5%	42.7	94.5%
relay - sink	43	99.5%	64.3	97.5%
Minstrel-HT				
source - relay	43.7	92.1%	41.5	91.9%
relay - sink	38.9	92.1%	85.6	98.6%

One observes that when the distance between the source and the relay is 82m, the transmission rate and the success ratio are similar to the ones obtained when the distance is 65m. However, the second link has a better quality which results in higher transmission rates. This increase of transmission rate has also an impact on the medium occupancy. Indeed, even if the number of sent frames on the second link is higher with a distance of 82m than with 65m, the radio medium is less used by the relay when $d = 82$. This results in more radio accesses for the source that can send more frames when $d = 82$ compared to $d = 65$ although the radio conditions are similar on the first link. For instance, the source with Iwl-Mvm-Rs sends 61900 frames when $d = 82$ whereas it sends 53535 frames when $d = 65$. As a result, the end-to-end throughput at the sink is higher when $d = 82$m.

5 CONCLUSION

In this paper, we have described the rate adaptation algorithm Iwl-Mvm-Rs used by Intel WiFi interfaces. To obtain this algorithm, we have retro-engineered the code of the IwlWiFi driver. We have shown that the Iwl-Mvm-Rs algorithm consists of cycles that include two main parts: the MCS scaling algorithm that seeks to maximize the throughput by changing the MCS and the Column scaling that seeks to find a better parameter combination including the mode (legacy, SISO, MIMO), the guard interval and the antenna configuration. We have implemented the Iwl-Mvm-Rs algorithm in the ns-3 simulator. Thanks to this implementation, we have compared the Iwl-Mvm-Rs algorithm with the Minstrel-HT and IdealWifi algorithms provided in NS-3. Minstrel-HT is also used by real WiFi interfaces. The obtained results on the tested scenarios show that:

- Without fading and node mobility, Minstrel-HT and Iwl-Mvm-Rs perform similarly, as the asymmetry in the number of lost frames or frames in success taken into account in the test window are not adapted to high variable radio conditions. even if the rate adaptation mechanisms are different and do not lead to the same used transmission rates and success ratio. IdealWifi obtains limited performance due to its conservative and rigid behavior.
- Without fading and with mobility, Iwl-Mvm-Rs shows good results, compared to Minstrel-HT and IdealWifi, specifically when the throughput of the source is high. Iwl-Mvm-Rs is able to quickly adapt its transmission rate to the change of radio conditions due to the presence of walls contrary to Minstrel-HT and IdealWifi.
- With fading and whatever the mobility, the use of Iwl-Mvm-Rs gives lower performance than with Minstrel-HT and IdealWifi. The algorithm has difficulties to deal with the randomness introduced by the fading, due in parts to the small test windows on which it bases its decisions, as well as the asymmetry in the number of lost frames or successful frames needed to makes its decisions.

ACKNOWLEDGMENTS

The authors would like to thanks the Direction Générale de l'Armement (DGA) and the Fédération Informatique de Lyon (FIL) for their financial support.

REFERENCES

[1] 2016. IEEE Standard for Information technology - Telecommunications and information exchange between systems Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)* (Dec 2016), 1–3534. https://doi.org/10.1109/IEEESTD.2016.7786995

[2] Linux Kernel Contributors. [n.d.]. Intel IwlWiFi Driver ; Linux Kernel Source Tree. https://github.com/torvalds/linux/tree/master/drivers/net/wireless/intel/iwlwifi.

[3] Lara B. Deek, Eduard Garcia Villegas, Elizabeth M. Belding, Sung-Ju Lee, and Kevin C. Almeroth. 2014. Intelligent Channel Bonding in 802.11n WLANs. *IEEE Trans. Mob. Comput.* 13, 6 (2014), 1242–1255. https://doi.org/10.1109/TMC.2013.73

[4] S. Derek. [n.d.]. Minstrel. http://madwifi-project.org/browser/madwifi/trunk/ath_rate/minstrel/minstrel.txt.

[5] Kai-Ten Feng, Po-Tai Lin, and Wen-Jiunn Liu. 2010. Frame-Aggregated Link Adaptation Protocol for Next Generation Wireless Local Area Networks. *EURASIP J. Wireless Comm. and Networking* 2010 (2010). https://doi.org/10.1155/2010/164651

[6] F. Fietkau. [n.d.]. Minstrel HT: New rate control module for 802.11n. https://lwn.net/Articles/376765/.

[7] R. Grünblatt, I. Guérin-Lassous, and O. Simonin. [n.d.]. Source Code of the Iwl-Mvm-Rs implementation in the NS-3 simulator. https://github.com/rgrunbla/ns-3-iwl-mvm-rs.

[8] Raja Karmakar, Samiran Chattopadhyay, and Sandip Chakraborty. 2017. IEEE 802.11ac Link Adaptation Under Mobility. In *42nd IEEE Conference on Local Computer Networks, LCN 2017, Singapore, October 9-12, 2017*. 392–400. https://doi.org/10.1109/LCN.2017.90

[9] Lito Kriara and Mahesh K. Marina. 2015. SampleLite: A Hybrid Approach to 802.11n Link Adaptation. *Computer Communication Review* 45, 2 (2015), 4–13. https://doi.org/10.1145/2766330.2766332

[10] Duy Nguyen and J. J. Garcia-Luna-Aceves. 2011. A practical approach to rate adaptation for multi-antenna systems. In *Proceedings of the 19th annual IEEE International Conference on Network Protocols, ICNP 2011, Vancouver, BC, Canada, October 17-20, 2011*. 331–340. https://doi.org/10.1109/ICNP.2011.6089072

[11] Ioannis Pefkianakis, Yun Hu, Starsky H. Y. Wong, Hao Yang, and Songwu Lu. 2010. MIMO rate adaptation in 802.11n wireless networks. In *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking, MOBICOM 2010, Chicago, Illinois, USA, September 20-24, 2010*. 257–268. https://doi.org/10.1145/1859995.1860025

[12] Sanjib Sur, Ioannis Pefkianakis, Xinyu Zhang, and Kyu-Han Kim. 2016. Practical MU-MIMO user selection on 802.11ac commodity networks. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, MobiCom 2016, New York City, NY, USA, October 3-7, 2016*. 122–134. https://doi.org/10.1145/2973750.2973758

[13] Sanjib Sur, Ioannis Pefkianakis, Xinyu Zhang, and Kyu-Han Kim. 2016. Practical MU-MIMO User Selection on 802.11Ac Commodity Networks. In *Proceedings of the 22Nd Annual International Conference on Mobile Computing and Networking (MobiCom '16)*. ACM, New York, NY, USA, 122–134. https://doi.org/10.1145/2973750.2973758

[14] Iñaki Ucar, Carlos Donato, Pablo Serrano, Andres Garcia-Saavedra, Arturo Azcorra, and Albert Banchs. 2016. Revisiting 802.11 Rate Adaptation from Energy Consumption's Perspective. In *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '16)*. ACM, New York, NY, USA, 27–34. https://doi.org/10.1145/2988287.2989149